2Х

УДК 519.6

О.М. Хіміч, О.В. Попов, В.В. Полянко
Інститут кібернетики ім. В.М. Глушкова НАН України, м. Київ, Україна
dept150@insyg.kiev.ua

Технології високопродуктивних обчислень для дослідження та розв'язування задач розрахунку міцності конструкцій

Розглядаються технології високопродуктивних обчислень для дослідження і розв'язування задач (статичних лінійних і нелінійних та динамічних лінійних) розрахунку міцності конструкцій. Описано паралельні алгоритми формування, дослідження та розв'язування дискретних задач, які одержуються в результаті дискретизації методом скінченних елементів вихідних задач, а також реалізація запропонованих технологій в програмному комплексі розв'язування задач розрахунку міцності будівельних конструкцій.

Вступ

При проектуванні об'єктів різного призначення часто виникає необхідність в проведенні розрахунків міцності конструкції таких об'єктів або окремих її елементів. В першу чергу задачі розрахунку міцності конструкцій виникають у промисловому та цивільному будівництві – при розрахунках споруд в цілому або їх окремих конструктивних елементів, а також в різних галузях машинобудування – суднобудуванні, авіабудуванні, ракетобудуванні, моторобудуванні тощо.

Зростаючі вимоги до якості проектних рішень, застосування нових конструктивних матеріалів викликає необхідність у розв'язуванні якісно нових задач. Зростає потреба в нових методах і підходах, пов'язаних з побудовою і дослідженням коректних комп'ютерних моделей, які адекватно відображають реальну поведінку конструкцій. Зростають вимоги до достовірності одержуваних комп'ютерних результатів. Ці чинники призводять до того, що істотно зростає обсяг оброблюваної інформації. Також великий обсяг інформації необхідно обробляти при виконанні розрахунків унікальних конструкцій. Тому виникає потреба у використанні високопродуктивних обчислень, що у свою чергу викликає зростання вимог до обчислювальних ресурсів.

З іншого боку, на сучасному етапі розвитку обчислювальної техніки відбувається перехід від комп'ютерів з одним одноядерним процесором, можливості підвищення продуктивності яких вже вичерпано, до комп'ютерів з багатоядерними процесорами, яких теж може бути багато. Використання багатоядерних і багатопроцесорних комп'ютерних систем є необхідною передумовою створення технологій високопродуктивних обчислень. Але ефективність таких технологій визначається переважно відповідністю між програмно-алгоритмічним забезпеченням і технічними засобами багатоядерного і багатопроцесорного комп'ютера. Адже найвищої продуктивності можна досягти, використовуючи алгоритми і програми з паралельною організацією обчислень. Тому виникає потреба як в адаптуванні для багатопроцесорних комп'ютерів раніше створеного програмного забезпечення комп'ютерів з одним одноядерним процесором, так і в створенні якісно нового програмного забезпечення, яке використовує паралельні алгоритми дослідження та розв'язування задач.

Метою цієї роботи є впровадження високопродуктивних обчислень в існуючі програмні засоби для комп'ютерів з одним одноядерним процесором.

Так, для проведення розрахунків міцності конструкцій створено багато програмних засобів, як універсального призначення, так і орієнтованих на певну галузь (NASTRAN, MARC, ANSYS тощо). Наприклад, для розрахунку міцності будівельних конструкцій розроблено вітчизняний програмний комплекс (ПК) ЛІРА [1]. Настроївши його інтерфейс для відповідної предметної області, ПК ЛІРА можна використовувати для розрахунків в інших галузях.

На базі препроцесора і постпроцесора ПК ЛІРА та інтелектуальних паралельних програм бібліотеки Inparlib [2], [3] розроблено програмний комплекс, який реалізує технології високопродуктивних обчислень для дослідження і розв’язування задач розрахунку міцності будівельних конструкцій на комп’ютерах з паралельною організацією обчислень, зокрема, на інтелектуальних паралельних комп’ютерах Інпарком [4].

Математичні постановки задач

Математично задачі розрахунку міцності конструкцій з використанням принципу можливих зміщень можуть бути поставлені у вигляді варіаційних задач: необхідно знайти вектор-функцію $u\in U_0$, яка для будь-якої вектор-функції $v\in U_0$ (для будь-якого можливого зміщення) задовольняє одній з інтегральних тотожностей

– для статичної задачі

$$a(u,v) = l(f,v); \tag{1}$$

– для динамічної задачі

$$a(u,v) + b(u'',v) + c(u',v) = l(f,v), \qquad u(t_0) = u^{(0)}, \quad u'(t_0) = u^{(1)}; \tag{2}$$

– для задачі на власні коливання

$$a(u,v) = \lambda b(u,v), \tag{3}$$

де U_0 – нескінченновимірний функціональний простір можливих переміщень, симетричні білінійні функціонали $a(u,v) = \int\limits_V \varepsilon^T(v)\,\tau(u)\,dV$, $b(u'',v)$, $c(u',v)$ пропорційні відповідно потенційній, кінетичній енергіям деформації і енергії гальмування, а лінійний функціонал $l(f,v)$ пропорційний роботі прикладених (зовнішніх) зусиль при навантаженні (тут $u^T = (u_x, u_y, u_z)$ – вектор-функція зміщень, через u' позначено її першу похідну за часом, а через u'' – другу, $\varepsilon^T = [\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \varepsilon_{xy}, \varepsilon_{yz}, \varepsilon_{zx}]$ – вектор деформацій, $\tau^T = [\tau_{xx}, \tau_{yy}, \tau_{zz}, \tau_{xy}, \tau_{yz}, \tau_{zx}]$ – вектор напруг, V – об’єм, S – поверхня тіла).

Вектори напруг τ і деформацій ε зв’язані такою рівністю

$$\tau = C\varepsilon, \tag{4}$$

де C – узагальнена матриця пружності. У випадку нелінійної задачі елементи цієї матриці C залежать від деформацій, а отже, і від зміщень, на відміну від випадку лінійної задачі, коли ці елементи не залежать від шуканої функції u.

Теоретичною основою більшості програмних засобів для розрахунку міцності конструкцій є метод скінченних елементів (МСЕ), реалізований у формі зміщень. Вибір саме цієї форми пояснюється простотою її алгоритмізації і фізичної інтерпретації, наявністю єдиних методів побудови матриць жорсткості і векторів навантажень для різних типів скінченних елементів, можливістю врахування довільних граничних умов і складної геометрії конструкції, що розраховується.

Дискретизація варіаційної задачі методом скінченних елементів, який є типовим представником проекційних методів, полягає в заміні нескінченновимірного простору допустимих функцій U_0 його скінченновимірним підпростором U_0^h. Вектор-функції з підпростору U_0^h можуть бути представлені у вигляді лінійної комбінації базисних

вектор-функцій, які задовольняють головним (кінетичним) граничним умовам,

$$u^h(\mathrm{X}) = \sum_{i=1}^{n} x_i \varphi_i(\mathrm{X}), \qquad (5)$$

де φ_i ($i = 1, 2, \ldots, n$) – згаданий вище базис U_0^{h}, звичайно кусково-поліноміальний.

Тоді білінійні і лінійний функціонали з (1) – (3) в скінченновимірному просторі U_0^{h} у випадку вихідної лінійної задачі можна записати у вигляді відповідно білінійних і лінійної форм коефіцієнтів x_i в представленні (5), які надалі називатимемо вузловими параметрами:

$$a(u^h,v^h) \equiv y^{\mathrm{T}}Ax, \quad b(u^h,v^h) \equiv y^{\mathrm{T}}Bx, \quad c(u^h,v^h) \equiv y^{\mathrm{T}}Cx, \quad l(f,v^h) \equiv y^{\mathrm{T}}b, \qquad (6)$$

де x і y – вектори вузлових параметрів відповідно функцій u^h і v^h, а елементи матриць жорсткості (A), мас (B), демпфування (C) і вектора навантажень (b) обчислюються за формулами ($i, j = 1, 2, \ldots, n$)

$$a_{ij} = a(\varphi_i, \varphi_j), \quad b_{ij} = b(\varphi_i, \varphi_j), \quad c_{ij} = c(\varphi_i, \varphi_j), \quad b_i = l(f, \varphi_i). \qquad (7)$$

Таким чином, враховуючи в (6) довільність вектора y, отримуємо відповідні дискретні задачі [5]:

– для статичної задачі (1) – систему лінійних алгебраїчних рівнянь (СЛАР)

$$Ax = b; \qquad (8)$$

– для динамічної задачі (2) – задачу з початковими умовами

$$Bx''(t) + Cx'(t) + Ax(t) = b(t), \quad x(t_0) = x^{(0)}, \quad x'(t_0) = x^{(1)}; \qquad (9)$$

– для задачі на власні коливання (3) – узагальнену алгебраїчну проблему власних значень

$$Ax = \lambda^{\mathrm{h}}Bx. \qquad (10)$$

Але для випадку нелінійної задачі дискретну задачу так отримати не можна. Задачу спершу треба лінеаризувати. Як правило, лінеаризація полягає в тому, що будується ітераційний процес: вибирається початкове наближення C_0 для узагальненої матриці пружності C з (4), тоді (у випадку стаціонарної задачі) отримуємо лінійну задачу $a_1(u^h,v^h) = l(f,v^h)$, розв'язок якої u_1^h використовується для обчислення наступного наближення C_1 узагальненої матриці пружності. Це наближення використовується для побудови наступної лінійної задачі $a_2(u^h,v^h) = l(f,v^h)$ і т.д. доти, доки не буде виконана умова збіжності, наприклад, така

$$\frac{\left\| u_t^h - u_{t-1}^h \right\|}{\left\| u_t^h \right\|} < \varepsilon .$$

Таким чином, ітераційний процес полягає в знаходженні послідовності функцій $u_t^h \in U_0^h$, $t = 1, 2, \ldots$, кожна з яких при довільній вектор-функції $v^h \in U_0^h$ задовольняє інтегральній тотожності $a_t(u^h,v^h) = l(f,v^h)$ з білінійним і лінійним функціоналами. Кожна з цих лінійних варіаційних задач легко зводиться до СЛАР виду (8) $A_t x = b$, $t = 1, 2, \ldots$. В деяких випадках нелінійність моделюється змінними навантаженнями, і тоді ітераційний процес будується так, що від ітерації до ітерації змінюється тільки права частина системи (8), а матриця залишається постійною, тобто послідовно розв'язуються задачі $Ax = b_t$, $t = 1, 2, \ldots$.

Формування і розв'язування дискретних задач

Враховуючи, що базисні функції підпростору МСЕ U_0^h можна вибрати так, щоб вони були відмінними від нуля лише на декількох скінченних елементах, структура матриць з (6) є в загальному випадку *розрідженою* і визначається нумерацією вузлових параметрів, яка у свою чергу залежить від нумерації вузлів скінченноелементної сітки. Тобто матриці задач (8) – (10) є стрічковими, профільними, блочно-діагональ-

ними з обрамленням тощо. Крім того, з точки зору комп’ютерного розв’язування цих задач істотне значення має те, що матриці дискретних задач є *симетричними*, *додатно означеними* або *додатно напівозначеними*, а також, що порядок цих матриць складає $O(10^5) – O(10^7)$.

Формування дискретних задач. Для побудови простору МСЕ U_0^h вихідна область розбивається на підобласті-елементи. На кожному елементі функції з простору МСЕ є поліномами заданого степеня, коефіцієнти яких визначаються через вузлові параметри – значення функцій (а у ряді випадків і їх похідних) у вузлах елемента (точки у вершинах, на сторонах, гранях або усередині елемента). Для вузлових параметрів і базисних функцій справедливі такі співвідношення

$$L_j(\varphi_i) = \delta_{ij}, \tag{11}$$

де δ_{ij} – символ Кронекера, а результатом операції $Lj(\varphi_i)$ є значення компоненти вектор-функції φ_i для вузлового параметра Lj.

Важливою перевагою МСЕ, враховуючи (5) і (11), є те, що елементи матриць і векторів правих частин задач (8) – (10) отримують підсумовуванням відповідних елементів відповідних матриць і векторів навантажень, побудованих для окремих скінченних елементів. Така властивість матриць і векторів дискретних задач МСЕ дозволяє ефективно розпаралелювати процес формування цих матриць і векторів [6]. При цьому можливо розпаралелювати тільки обчислення та розподіл між паралельними процесами відповідно до вимог методу, який використовуватиметься для розв’язування дискретної задачі, елементів глобальних матриць і векторів дискретних задач, використовуючи обчислені раніше матриці і вектори окремих скінченних елементів. Або розпаралелювати також обчислення цих матриць і векторів окремих скінченних елементів.

З метою мінімізації кількості арифметичних операцій при розв’язуванні дискретних задач, а також необхідної оперативної і зовнішньої пам’яті, враховуючи розріджену структуру матриць задач (8) – (10), в більшості випадків проводиться переупорядковування невідомих задач. Залежно від критерію мінімізації – ширини стрічки матриці, її профілю, загальної кількості арифметичних операцій при трикутному розвиненні матриці і т.д. – використовуються різні методи переупорядковування, наприклад, зворотний алгоритм Катхілла-Маккі, фактор дерев, мінімального степеня [7]. Конкретних рекомендацій для вибору методу впорядкування дати не можна, тому що ефективність того або іншого алгоритму суттєво залежить від первинної структури конкретної матриці.

Розв’язування систем лінійних алгебраїчних рівнянь. Для розв’язування СЛАР (8) – вихідної або переупорядкованої – на MIMD-комп’ютері можуть бути застосовані паралельні алгоритми для дослідження і розв’язування СЛАР з симетричними матрицями, які використовують LDL^{T}-розвинення (або U^TDU-розвинення) методу Холецького [3], [8]. Залежно від ширини і заповнення стрічки матриці використовуються:

– блочні паралельні алгоритми для вузьких стрічкових матриць і блочно-діагональних матриць з обрамленням;

– одновимірні блочні циклічні паралельні алгоритми для дослідження і розв’язування СЛАР із стрічковими або профільними матрицями;

– одновимірний блочний циклічний паралельний алгоритм (U^TDU-розвинення у формі зовнішніх добутків) для дослідження і розв’язування СЛАР з матрицями так званої хмарочосної структури (у яких не беруть участь в обчисленнях внутрішні елементи профілю матриці, які залишаються нульовими в процесі розвинення матриці).

Розв’язування задачі з початковими умовами. В напівдискретному МСЕ наближений розв’язок шукається у вигляді (5), де коефіцієнти x_i є функціями часу t. В результаті отримуємо систему (9) звичайних диференціальних рівнянь другого по-

рядку з початковими умовами, де: $x(t)$, $x^{(0)}$, $x^{(1)}$ – вектори з елементами $x_i(t)$, $x_i^{(0)} = L_i(u^{(0)}), \quad x_i^{(1)} = L_i(u^{(1)})$.

В переважній більшості програмних засобів розрахунку міцності конструкцій система (9) розв'язується методом розвинення за формами власних коливань. Якщо λ_k^h, z_k^h $(z_k^{h\mathsf{T}} B z_k^h = 1,\ k = 1, 2, \ldots, n)$ – розв'язок алгебраїчної проблеми власних значень (10), то, вважаючи в (9) $x(t) = \sum_{k=1}^{n} y_k(t) z_k^h$, отримаємо (при B-ортогональності векторів z_k^h та деяких припущеннях щодо матриці демпфування C) систему, яка розпадається на незалежні відносно $y_i(t)$ рівняння:

$$\begin{aligned} & y_k''(t) + 2\xi_k \omega_k y_k'(t) + \omega_k^2 y_k(t) = P_k(t), \quad t > t_0, \\ & y_k(t_0) = y_k^{(0)}, \quad y_k'(t_0) = y_k^{(1)}, \end{aligned} \tag{12}$$

де $\omega_i = \lambda_i^{-0,5}$, $0 < \xi_k < 1$, $P_k(t) = z_k^{h\mathsf{T}} b(t)$, $y_k^{(0)} = x^{(0)\mathsf{T}} B z_k^h$, $y_k^{(1)} = x^{(1)\mathsf{T}} B z_k^h$, $k = 1, 2, \ldots, n$. Розв'язки задач (12) мають вигляд:

$$y_k(t) = e^{-\xi_k \omega_k t} \left(\frac{y_k^{(1)} + y_k^{(0)} \xi_k \omega_k}{w_k} \sin \overline{\omega}_k t + y_k^{(0)} \cos \overline{\omega}_k t \right) + \frac{1}{\overline{\omega}_k} \int_{t_0}^{t} P_k(\tau)^{-\xi_k \omega_k (t-\tau)} \sin \overline{\omega}_k (t-\tau) d\tau,$$

де $\overline{\omega}_k = \omega_k \sqrt{1 - \xi_k}$. Аналіз цих розв'язків свідчить про те, що істотний внесок в розв'язок $x(t)$ задачі (8) дають лише близько 10 – 20 складових, які відповідають мінімальним власним значенням.

Існує ряд випадків навантажень, коли можливе точне обчислення розв'язків задач (12), наприклад, для вітрового, сейсмічного або гармонійного навантажень. В решті випадків розв'язки $y_k(t)$ обчислюються чисельно, наприклад, методом скінченних різниць за схемою Ньюмарка. Також задачі Коші (12) можуть розв'язуватися методом Рунге – Кутта четвертого порядку, орієнтованим на розв'язування систем звичайних диференціальних рівнянь другого порядку.

Розв'язування узагальненої алгебраїчної проблеми власних значень. В задачі на власні коливання, а також, як зазначалось вище, при використанні методу розвинення за формами власних коливань, як правило, необхідно знайти лише невелику кількість власних форм (порівняно з порядком задачі), які відповідають мінімальним власним значенням. Тому відповідна часткова проблема власних значень (10) може розв'язуватися методом ітерацій на підпросторі, паралельні алгоритми якого описано в [3].

Для отримання наближеного розв'язку за допомогою цього методу важливо оптимізувати вибір розмірності підпростору, що ітерується, залежно від кількості необхідних власних пар. Важливо також з огляду на те, що матриця маси B задачі (10) часто є напівозначеною, так вибирати початковий підпростір, щоб матриці проекцій на підпростір були додатно означеними.

Слід також брати до уваги, що кількість ненульових елементів матриці мас B задачі (10), як правило, значно менша за кількість елементів матриці жорсткості A, які беруть участь в обчисленнях. Тому з точки зору підвищення ефективності паралельного алгоритму доцільно зберігати всі ненульові елементи матриці мас, не зважаючи на те, що вона є симетричною. Такий підхід значно спрощує паралельний алгоритм множення розрідженої симетричної матриці на прямокутну матрицю і зменшує кількість обмінів даними та обсяг цих даних.

При виборі паралельного алгоритму необхідно враховувати його ефективність. Коефіцієнти прискорення та ефективності паралельних алгоритмів, які згадано вище

та які можуть використовуватись в задачах розрахунку міцності будівельних конструкцій, подано в монографії [3] та роботі [8].

На вибір паралельних алгоритмів при розв'язуванні задач з розрідженими матрицями суттєво впливає також збалансованість завантаження паралельних процесів обчисленнями. Тому для кожної задачі має обиратися такий метод оптимізації структури матриць і такий алгоритм розв'язування задачі та його параметри, які забезпечують не тільки зменшення загальної кількості арифметичних операцій, але і мінімізацію часу очікування даних окремими процесами при обчисленнях.

Достовірність наближених розв'язків

При розрахунку міцності конструкції важливо гарантувати достовірність отримуваних розв'язків. Коли йдеться про достовірність розв'язку, то мається на увазі, що досліджується достовірність наближених розв'язків математичних задач (1) – (3). Повна похибка наближеного розв'язку враховує вплив похибок вихідних даних, похибок дискретизації і формування даних відповідної дискретної задачі (8), (9) або (10) та похибок комп'ютерного розв'язку цієї дискретної задачі.

Достовірність розв'язку прикладних задач з **наближеними вихідними даними** гарантується використанням стійких до збурення вихідних даних математичних моделей конструкцій, що розраховуються. Використання при дискретизації МСЕ теоретично обґрунтованих скінченних елементів, які задовольняють умовам **збіжності** і для яких отримано **оцінки похибок розв'язку**, є передумовою отримання достовірного наближеного розв'язку задачі. Для отримання достовірного розв'язку також важливо узгоджувати точність **квадратурних формул**, які використовуються при обчисленнях (7) елементів матриць (жорсткостей, мас, демпфування) і вектора навантажень, з точністю МСЕ, враховуючи при цьому поведінку підінтегральних функцій.

Похибка комп'ютерного розв'язку $\tilde{x}$ відносно точного розв'язку x відповідної дискретної задачі (8), (9) або (10) залежить від точності завдання вихідних даних, методу і алгоритму розв'язування цієї дискретної задачі, а також похибок заокруглення. Достовірність розв'язку також залежить від властивостей комп'ютерної задачі, які можуть суттєво відрізнятися від властивостей вихідної задачі. Наприклад, при розв'язування СЛАР важливу роль відіграє число обумовленості матриці, а при розв'язуванні часткової проблеми власних значень необхідно пересвідчитись, що всі обчислені власні значення є найменшими. Тому апріорне та апостеріорне дослідження математичних властивостей комп'ютерних версій дискретних задач (8), (9) або (10) є необхідною умовою отримання достовірного розв'язку. Комп'ютерні методи та паралельні алгоритми дослідження властивостей дискретних задач та їх розв'язків описано в монографії [3].

Після отримання розв'язку задачі необхідно провести апостеріорний аналіз результатів з точки зору задоволення тим гіпотезам, які були використані при побудові математичної моделі задачі, а також вимогам точності розв'язку. Якщо розв'язок задовольняє всім поставленим вимогам, то можна вважати, що цьому розв'язку можна довіряти і використовувати.

Технологія дослідження та розв'язування задач розрахунку міцності конструкцій на комп'ютерах з паралельною організацією обчислень

Основна ідея сучасного підходу – інтеграція в рамках єдиного програмного комплексу виконання наступних завдань:

– інтерактивна постановка задачі – створення скінченноелементної моделі конструкції;

– автоматичне формування даних дискретної задачі;
– автоматичне дослідження та розв'язування дискретної задачі;
– автоматична обробка результатів розв'язування дискретної задачі;
– інтерактивний аналіз та використання результатів розрахунку.

Паралельні обчислення варто використовувати для завдань, які виконуються автоматично, в першу чергу, для дослідження та розв'язування дискретних задач (тому що, як правило, це завдання вимагає найбільше часу).

Як наголошувалося вище для дослідження та розв'язування задач (1) – (3) на комп'ютерах з паралельною організацією обчислень створено програмний комплекс (ПК) розв'язування задач розрахунку міцності будівельних конструкцій [2], [3]. Цей ПК (надалі ПК ЛІРА-cluster) розроблено на базі ПК ЛІРА та інтелектуальних паралельних програм бібліотеки Inparlib [2], [3] для інтелектуальних паралельних комп'ютерів Інпарком. Після деякої адаптації (яка в більшості випадків полягає у перекомпіляції паралельних програм) ПК ЛІРА-cluster може бути встановлено на інших багатопроцесорних комп'ютерних системах.

При створенні цього програмного комплексу необхідно було взяти до уваги, що графічний інтерфейс пре- та постпроцесора ПК ЛІРА, як і в більшості аналогічних програмних комплексів, працює в середовищі ОС Windows, а програмні модулі, які виконують паралельні обчислення, – в ОС Linux (ця ОС використовується на більшості багатопроцесорних комп'ютерів). Тому при створенні комплексу використано те, що в склад багатопроцесорної робочої станції Інпарком входить host-система, на одному з комп'ютерів (графічній станції) якої може бути завантажено ОС Windows, в той час як на обчислювальні вузли – ОС Linux [2]. Як host-комп'ютер в ПК ЛІРА-cluster може використовуватись комп'ютер користувача, зокрема персональний. Такий підхід дозволяє поетапно переходити до використання паралельних обчислень при розрахунку міцності будівельних конструкцій, починаючи з найбільш ресурсоємного етапу – дослідження та розв'язування дискретних задач.

Розрахунок міцності конструкції за допомогою ПК ЛІРА-cluster проводиться наступним чином:

– на *графічній станції* Інпарком інтерактивно на мові предметної області, використовуючи *препроцесор* ПК ЛІРА, створюється або вводиться раніше створена скінченноелементна модель конструкції;

– на *графічній станції* автоматично *препроцесором* програмного комплексу формуються вихідні дані для формування (можливо, часткового) та розв'язування на обчислювальних вузлах відповідної дискретної задачі (8) або (10); ці дані записуються в бінарні файли, використовуючи прийняті в ПК ЛІРА формати даних;

– автоматично за допомогою спеціального інтерфейсу на *обчислювальні вузли* комп'ютера Інпарком завантажується *паралельна програма*, яка: 1) визначає алгоритм і параметри розв'язування задачі, 2) зчитує вихідні дані з файлів та формує розподілену між паралельними процесами відповідно до вибраного алгоритму розв'язування дискретну задачу, 3) досліджує і розв'язує дискретну задачу, використовуючи відповідні модулі бібліотеки Inparlib, 4) зберігає результати дослідження і розв'язування задачі у текстовому та бінарних файлах у форматах ПК ЛІРА для подальшої обробки і використання постпроцесором програмного комплексу;

– на *графічній станції* зчитуються з файлів результати розв'язування дискретної задачі та автоматично засобами ПК ЛІРА проводиться обробка отриманих результатів;

– на *графічній станції* інтерактивно на мові предметної області *постпроцесором* ПК ЛІРА проводиться аналіз та використання результатів розрахунку або результати зберігаються для подальшої обробки та використання на комп'ютері користувача.

Для розв’язування динамічних задач використовується метод розвинення за формами власних коливань. При цьому дослідження та розв’язування часткової узагальненої алгебраїчної проблеми власних значень виконується паралельно на обчислювальних вузлах, а розв’язування рівнянь (12) реалізується ПК ЛІРА, для чого в бінарних файлах потрібно зберігати обчислені власні значення та вектори, а також трикутне розвинення матриці жорсткостей.

У випадку розв’язування нелінійної задачі ітераційний процес побудовано так, що на кожній ітерації на обчислювальних вузлах для чергового наближення формується матриця жорсткостей задачі (8) та виконується її трикутне розвинення, а всі інші дії здійснюються ПК ЛІРА. Тобто для отримання розв’язку необхідно багаторазово звертатися до обчислювальних вузлів багатопроцесорного комп’ютера. В подальших версіях ПК ЛІРА-cluster планується повністю організувати ітераційний процес на обчислювальних вузлах, але це вимагає внесення значних змін у ПК ЛІРА.

Інтерфейс користувача ПК ЛІРА-cluster. В ПК ЛІРА-cluster використовується інтерфейс ПК ЛІРА, який доповнено двома вікнами. Тому робота користувача ПК ЛІРА-cluster практично не відрізняється від роботи з ПК ЛІРА.

Перше з цих вікон – стартове вікно ПК ЛІРА-cluster (рис. 1), з відкриття якого розпочинається робота з програмним комплексом. У ньому міститься деяка довідкова інформація про встановлений режим роботи програмного комплексу – кластерний (з паралельним процесором) або стандартний (з послідовним процесором) – та про стан багатопроцесорного комп’ютера (кластера) – дані про загальну та вільну кількість ядер. Також тут містяться кнопки, що дозволяють розпочати або закінчити сеанс роботи з програмним комплексом, змінити режим його роботи, відновити вікно вибору (зміни) кількості ядер (рис. 1).

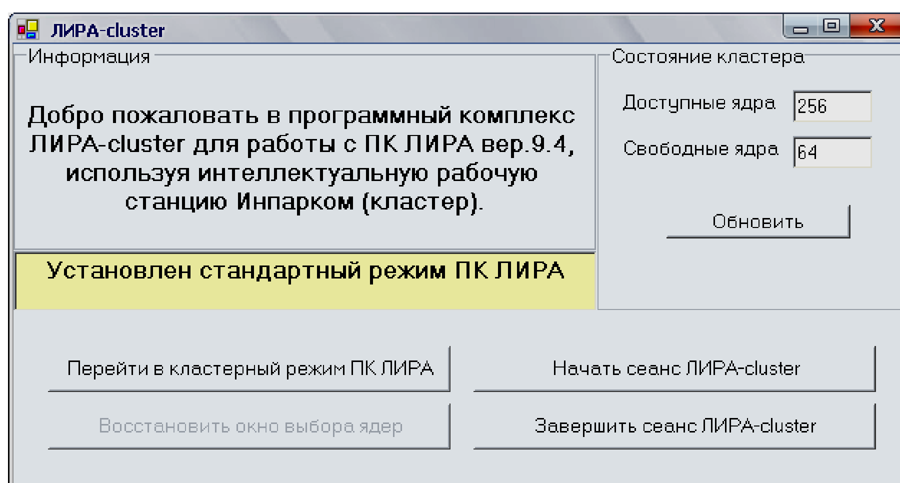


Рисунок 1 – Стартове вікно ПК ЛІРА-cluster

Друге вікно – вікно зміни кількості ядер – пов’язано безпосередньо з самим процесом паралельних обчислень. Так, коли закінчується формування даних для роботи паралельної частини ПК ЛІРА-cluster, відкривається це вікно. У ньому вказується загальний ресурс та кількість вільних ядер, яка визначається відповіддю на спеціальний запит кластеру безпосередньо відкриттям цього вікна. Також користувачу пропонується (поле «Рекомендовано») автоматично визначена, найкраща для даної задачі (з точки зору швидкодії) кількість ядер. У випадку, якщо користувача не задовольняє рекомендована кількість, він власноруч може задати інше число на свій розсуд. Введене число перевіряється на коректність, у тому числі й з урахуванням

реальних потужностей кластера. Для зручності у вікні розміщено кнопку оновлення кількості процесів. Якщо на даний момент кількість вільних ядер замала для даної задачі, користувач може очікувати на більше число, час від часу натискаючи кнопку оновлення. Коли користувач вважає, що задачу можна запускати на кластері, він натискає на кнопку «ОК», вікно зникає і розпочинається підготовка до запуску паралельної програми. Якщо ж з моменту появи вікна користувач не робив спроб змінити кількість ядер, через деякий час (інформація про який також дається у вікні) вікно закривається і розрахунок розпочинається автоматично.

Користувач також має можливість відмовитись від подальшої появи вікна вибору кількості ядер. В такому випадку це число визначається автоматично. Для відновлення режиму появи цього вікна необхідно використати відповідну кнопку стартового вікна ПК ЛІРА-cluster (яка активізується у момент відмови). Вікно вибору кількості ядер містить також поле для підказок, які подаються при наведенні курсора миші на відповідні елементи вікна.

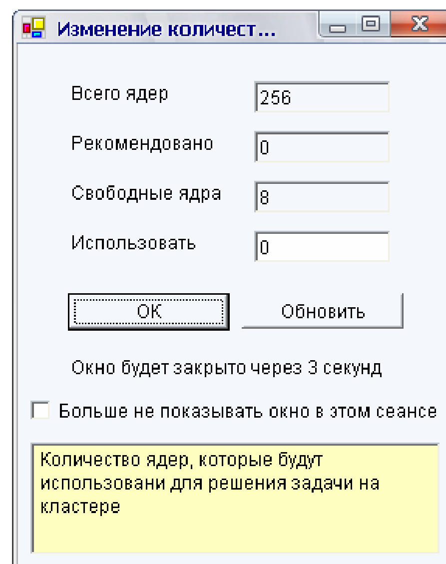


Рисунок 2 – Вікно вибору (зміни) кількості ядер

Перед запуском відповідної паралельної програми проводиться редагування команди запуску – встановлюється кількість процесів (яка дорівнює вибраній кількості ядер), якщо це число відмінне від наявного в команді запуску.

Користувач може працювати з ПК ЛІРА-cluster на інтелектуальному паралельному комп'ютері Інпарком – безпосередньо або використовуючи підключення через Інтернет до віддаленого робочого столу. Також передбачена можливість роботи з програмним комплексом, використовуючи як host-комп'ютер персональний комп'ютер користувача та Інтернет для зв'язку з багатопроцесорним комп'ютером. В останньому випадку перед обмінами даними між вказаними комп'ютерами проводиться їх архівація з метою зменшення обсягу даних, які пересилаються засобами каналів зв'язку.

Програмним комплексом ЛІРА-cluster на робочих станціях Інпарком проводилося розв'язування тестових задач. При цьому час розв'язування системи лінійних алгебраїчних рівнянь (8) зменшувався в 10 – 20 раз, а час розв'язування часткової алгебраїчної проблеми власних значень (10) – в 5 – 10 раз порівняно з розв'язуванням на персональному комп'ютері. Деякі приклади наведено в роботах [5], [9].

Висновки

Проведені авторами дослідження і апробація програмного комплексу для розв'язування задач розрахунку міцності будівельних конструкцій на робочих станціях Інпарком показали перспективність створення програмних засобів для розв'язування прикладних задач на паралельних комп'ютерах кластерного типу шляхом вбудовування паралельних програм для розв'язування окремих підзадач в наявні програмні комплекси в різних галузях науки і техніки. Як такі паралельні програми можуть використовуватися програми бібліотеки Inparlib або її аналогів на інших комп'ютерах з паралельною обробкою інформації.

Також є перспективним розширення кола задач, які розв'язуються за допомогою програмного комплексу ЛІРА-cluster на робочих станціях Інпарком, зокрема, більш глибокого розпаралелювання розв'язування нелінійних задач, а також обробки результатів розв'язування дискретних задач. Заслуговує на увагу також перенесення на обчислювальні вузли формування матриць (жорсткостей, мас) і векторів навантаження окремих скінченних елементів.

Література

1. [Электронный ресурс]. – Режим доступу : http://www.lira.com.ua.
2. Численное программное обеспечение интеллектуального MIMD-компьютера Инпарком / [Химич А.Н., Молчанов И.Н., Мова В.И. и др.]. – Київ : Наукова думка, 2007. – 216 с.
3. Параллельные алгоритмы решения задач вычислительной математики / [Химич А.Н., Молчанов И.Н., Попов А.В. и др.]. – К. : Наукова думка, 2008. – 248 с.
4. [Электронный ресурс]. – Режим доступу : http://www.inparcom.com.
5. Молчанов И.Н. Основы метода конечных элементов / И.Н. Молчанов, Л.Д. Николенко. – Київ : Наукова думка, 1984. – 267 с.
6. Химич А.Н. Решение задач расчета прочности конструкций на MIMD-компьютере / А.Н. Химич, В.В. Полянко, А.В. Попов, О.В. Рудич // Искусственный интеллект. – 2008. – № 3. – С. 750-760.
7. Джордж А. Численное решение больших разреженных систем уравнений / А. Джордж, Дж. Лю. – М. : Мир, 1984. – 333 с.
8. Хіміч О.М. Блочно-циклічні паралельні алгоритми методу Гауса розв'язування систем лінійних алгебраїчних рівнянь з розрідженими матрицями / О.М. Хіміч, В.В. Полянко // Вісник Львівського університету. Вип. 15. – 2009. – С. 109-116. – Серія «Прикладна математика та інформатика».
9. Хіміч О.М. Використання багатопроцесорних комп'ютерів для розв'язування задач розрахунку міцності конструкцій / О.М. Хіміч, В.В. Полянко, О.В. Попов, О.В. Рудич // Праці міжнародного симпозіуму «Питання оптимізації обчислень (ПОО-XXXV)». – Київ : Інститут кібернетики ім. В.М. Глушкова НАН України, 2009. – С. 382-387.

А.Н. Химич, А.В. Попов, В.В. Полянко

Технологии высокопроизводительных вычислений для исследования и решения задач расчета прочности конструкций

Рассматриваются технологии высокопроизводительных вычислений для исследования и решения задач (статических линейных и нелинейных и динамических линейных) расчета прочности конструкций. Описаны параллельные алгоритмы формирования, исследования и решения систем разрешающих уравнений, получаемых при дискретизации методом конечных элементов названных задач, а также реализация предложенных технологий в программном комплексе решения задач расчета прочности строительных конструкций.

A.N. Khimich, A.V. Popov, V.V. Polyanko

Technologies of High-performance Computations for the Investigating and Solving of Problems on the Strength Analysis of Constructions

Technologies of high-performance computations for investigating and solving of problems (static linear and non-linear, dynamical linear) on the strength analysis of constructions are dealt with. Parallel algorithms for the forming, investigating and solving of systems of resolving equations obtained during the finite-element discretization of the listed problems as well as implementation of technologies proposed in the program complex for the solving of problems on the strength analysis of constructions are described.

Стаття надійшла до редакції 30.06.2010.